

Applicant: Anderson	
Application No.: 10/699,486	Group Art Unit: 2165
Filed: 10/31/2003	Page 9 of 13
Title: TEXTUAL FILESYSTEM INTERFACE METHOD AND APPARATUS	Examiner: Mark A. Radtke
Attorney Docket No.: 200207252-1	

#### REMARKS

Applicant wishes to thank the Examiner for withdrawing rejected claim 15 under 35 USC 112 first paragraph, in the office action mailed November 1, 2006. In the same Office Action, however, the Examiner has maintained rejection of claims 1-24, under 35 USC 102(e) in view of U.S. Patent 6,856,993 by Verma et al. (hereinafter Verma). The Examiner also rejected claim 16 under 35 USC 103(a) over Verma in view of "On Optimistic Methods for Concurrency Control" by Kung et al., ACM Transactions on Database Systems (TODS), vol. 6, issue 2, pages 213-226, Published June 1981 (hereinafter Kung).

Applicant has amended the language of several claims to clarify aspects of the invention as claimed. Accordingly, Applicant believes the claim language as provided should make the various patentable aspects of the present invention more clear.

Accordingly, it should be clear that Verma does not anticipate or even suggest aspects of the instant invention as alleged under 35 USC 102(e). Verma describes an overview of many functions related to the operation of a transactional filesystem (Abstract; Col. 6-10; Col. 11, lines 1-16) The transactional filesystem described in Verma attempts to manage data stored in a file system when files may be requested or accessed by multiple different processes at or about the same time. Verma provides details on performing typical filesystems operations as a sequence of transactions through a transaction context object (Col. 9, lines 1-17; Col. 26, lines 38-63).

Applications interface with the transactional filesystem of Verma using various application programming interfaces (API) (Col. 6, lines 31-44) and proprietary servers and proxies (Col. 6, lines 56-67). These servers and proxy services serve as transaction coordinators (Col. 7, lines 1-14) and

Applicant: Anderson	
Application No.: 10/699,486	Group Art Unit: 2165
Filed: 10/31/2003	Page 10 of 13
Title: TEXTUAL FILESYSTEM INTERFACE METHOD AND APPARATUS	Examiner: Mark A. Radtke
Attorney Docket No.: 200207252-1	

have integrated protocols to initiate transactions and later either commit or abort the modification (Col. 6, lines 56-67). For example, Verma describes using transaction coordinators and a two-phase commit operation within the transactional filesystem (Col. 7, lines 60-65).

Indeed, Verma describes a transactional filesystem however it is only one example of such system and does describe or contemplate aspects of the present invention as presently claimed. For a proper § 102 rejection a single prior art reference must have every element of the claim rejected. See *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 U.S.P.Q.2D (BNA) 1913, 1920 (Fed. Cir.), cert. denied, 493 U.S. 853, 107 L. Ed. 2d 112, 110 S. Ct. 154 (1989) (explaining that an invention is anticipated if every element of the claimed invention, including all claim limitations, is shown in a single prior art reference). See *Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631, 2 U.S.P.Q.2D (BNA) 1051, 1053 (Fed. Cir. 1987) (explaining that a prior art reference anticipates a claim only if the reference discloses, either expressly or inherently, every limitation of the claim). See *Kloster Speedsteel AB v. Crucible, Inc.*, 793 F.2d 1565, 1571, 230 U.S.P.Q. (BNA) 81, 84 (Fed. Cir. 1986) ("Absence from the reference of any claimed element negates anticipation.")

Unfortunately, Verma does not disclose expressly or even inherently every element of the claims rejected. For example, Verma does not describe or suggest "duplicating the filesystem within a pseudo-filesystem" as recited in amended claim 1. The 'copyFile' command in Verma deals with a conventional copying of a file within the filesystem for an application or user but does not have the ability to copy a complete filesystem.

Applicant respectfully submits that Verma also stipulates that this is for applications and not internal management/operation of a transactional filesystems (Col. 10, lines 20-22: "In addition, it

Applicant: Anderson	
Application No.: 10/699,486	Group Art Unit: 2165
Filed: 10/31/2003	Page 11 of 13
Title: TEXTUAL FILESYSTEM INTERFACE METHOD AND APPARATUS	Examiner: Mark A. Radtke
Attorney Docket No.: 200207252-1	

allows *applications* (emphasis added) to do file-system operations like deleteFile and copyFile which do not have a transacted mode bit"). There is simply no mention of creating a pseudo-filesystem in Verma let alone interacting with the pseudo-filesystem as part of the transactional filesystem solution in Verma. For at least this reason, claim 1 as filed remains patentable over the cited art.

Further, Verma also does not describe, "creating a control text file that provides a textual filesystem interface and receives text-based commands to operate on the pseudo-filesystem" as recited in amended claim 1. Indeed, Verma specifically uses a conventional application programming interface (API) that requiring compilation and integration into a specific application (Col. 6, lines 31-55; Col. 7, lines 1-28.) In general, the filesystem in Verma also uses object-oriented technology and interface mechanisms and consequently teaches away from using a text-based interface. (Col. 7, lines 1-67; Col. 8, lines 1-67; Col. 9, lines 1-25.)

Applicant submits that the "log" files in Verma do not receive commands for controlling the transactional file system but instead receive the results of certain operations and/or errors (Col. 19, lines 60-67; Col. 20, lines 1-5). Indeed, the "log" files are not specified as being text files and would instead need to be objects as the rest of file system is designed around an object-oriented construct. (Col. 14, lines 50-67; Col. 15-16; Col. 17, lines 1-8). Logging service 74 in Verma aggregates these object-oriented logs from the filesystem to allow for multiple-level recovery system. (Col. 17, lines 44-53.) Neither the logs or the multiple-level recovery system provide "control text file" as recited in claim 1 that receives text based commands.

Applicant: Anderson	
Application No.: 10/699,486	Group Art Unit: 2165
Filed: 10/31/2003	Page 12 of 13
Title: TEXTUAL FILESYSTEM INTERFACE METHOD AND APPARATUS	Examiner: Mark A. Radtke
Attorney Docket No.: 200207252-1	

The control text file is a special text based interface to the transactional filesystem and requires no special or proprietary APIs as described in Verma. To interact with the transactional filesystem of claim 1, another filesystem or application need only send text data and text commands to the control text file. This "open interface" to the transactional filesystem provides many advantages in terms of accessibility and interoperability over the proprietary APIs described and suggested by Verma (Col. 6, lines 31-67). For example, the control text file as recited in claim 1 may allow different heterogeneous file systems and operating environments to more directly interact with a transactional filesystem since no APIs, proprietary interfaces or special server software are required.

Dependant claims 2 and 4-11 are not only patentably distinct on their own but also by virtue of their dependence on claim 1 as filed.

Similarly, independent claim 12, independent claim 19, independent claim 20, independent claim 21, independent claim 22, independent claim 23 and independent claim 24 remain patentably distinct over the cited art for at least the same reasons as described previously with respect to claim 1. Moreover, not only are dependant claims 13-18 patentably distinct on their own, they also are in condition for allowance by virtue of their dependence on independent claim 12.

The Examiner also rejected claim 16 under 35 USC 103(a) as being obvious over Verma in view of Kung. Applicant respectfully submit that Kung concerns optimistic methods for concurrency control however it does not describe or suggest, "using optimistic concurrency control (OCC) to control pending read and write operations to the pseudo-filesystem" as recited in claim 16. Because neither Kung or Verma teach or suggest this aspect of the invention, claim 16 remains patentably

MAR 01 2007

Applicant: Anderson

Application No.: 10/699,486

Filed: 10/31/2003

Title: TEXTUAL FILESYSTEM INTERFACE  
METHOD AND APPARATUS

Attorney Docket No.: 200207252-1

Group Art Unit:

2165

Page 13 of 13

Examiner:

Mark A. Radtke

distinct. Consequently, claim 16 not only remains independently patentable but also by virtue of its dependency on allowable claim 12.

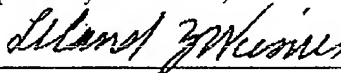
Applicant has made a diligent effort to place the claims in condition for allowance. In view of the remarks above, Applicant respectfully requests withdrawal of the rejections and allow claims 1-24.

In the event there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Leland Wiesner, Applicant's Attorney at (650) 853-1113 so that such issues may be resolved as expeditiously as possible.

03/01/2007  
Date

Wiesner and Associates  
366 Cambridge Avenue  
Palo Alto, California 94306  
Tel. (650) 853-1113

Respectfully Submitted,



Leland Wiesner

Attorney/Agent for Applicant Reg. No. 39424